

# Transfer Function Bounds for Partial-Unit-Memory Convolutional Codes Based on Reduced State Diagram

P. J. Lee

Communication Systems Research Section

*The performance of a coding system consisting of a convolutional encoder and a Viterbi decoder can be analytically found by the well-known transfer function bounding technique. For the partial-unit-memory byte-oriented convolutional encoder with  $m_o$  binary memory cells and  $k_o$  ( $> m_o$ ) inputs, a state diagram of  $2^{k_o}$  states has been used for the transfer function bound. In this article, it is shown that a reduced state diagram of  $(2^{m_o} + 1)$  states can be used for easy evaluation of transfer function bounds for partial-unit-memory codes.*

## I. Introduction

A class of convolutional codes called unit-memory byte-oriented convolutional codes (UM codes) was introduced by Lee (Ref. 1) and Lauer (Ref. 2). The encoder structure of an  $(m_o, k_o/n_o)$  UM code is shown in Fig. 1, where  $m_o$  is the number of binary memory cells,  $k_o$  is the number of inputs, and  $n_o$  is the number of outputs. The code rate  $r$  of this UM code is  $k_o/n_o$ . UM codes with  $m_o < k_o$  are called partial-UM codes (PUM codes) (Ref. 2). For consistency in terminology, we will call UM codes with  $m_o = k_o$ , full-UM codes (FUM codes).

In Ref. 1, it was shown that FUM codes are superior to conventional convolutional codes in the sense of having larger free distances for given pairs of  $m_o$  and  $r$ . In Ref. 2, PUM codes are shown to be even better than FUM codes in the same sense. Also, as inner codes for the concatenated coding systems with Reed-Solomon (RS) outer codes, FUM codes were shown to

be better than conventional convolutional codes, due to their byte-oriented natures (Refs. 1 and 3). For example, it was shown (Ref. 3) that, with a 6-bit RS outer code and convolutional inner codes with  $m_o = 6$  and  $r = 1/3$ , the use of FUM code can save 0.3 dB in required signal-to-noise ratio over the use of conventional convolutional code for a given performance. For the same applications, we expect that PUM codes will be more useful since we can employ larger symbol size RS codes with inner codes of small complexities (e.g., 8-bit RS code and  $(5, 8/n_o)$  PUM code).

The performance of a coding system employing a convolutional encoder and a Viterbi decoder can be analytically evaluated by the transfer function bound (TFB) based on the corresponding state diagram (SD) for that code (Refs. 4 and 5). However, for an  $(m_o, k_o/n_o)$  PUM code, the TFB should be evaluated from a SD of  $2^{k_o}$  states, and hence the inversion of a  $(2^{k_o} - 1) \times (2^{k_o} - 1)$  matrix is required. The purpose of this

report is to present a reduced state diagram of  $(2^{m_o} + 1)$  states to reduce the required computational burden for the evaluation of TFB.

## II. Preliminaries

This section briefly reviews the encoder structure and the state diagram (SD) for the evaluation of transfer function bound (TFB) for FUM codes. Readers who are not familiar with TFB are referred to Ref. 4 or Ref. 5.

The encoder structure of an  $(m_o, k_o/n_o)$  UM code is shown in Fig. 1. The  $(k_o + m_o)$ -input to  $n_o$ -output connection box including  $n_o \bmod 2$  adders is often represented by an  $n_o \times (k_o + m_o)$  binary matrix  $G$ , called a code generator matrix. The  $n$ th bit in the  $t$ th output vector,  $y_n^t$ ,  $n = 1, 2, \dots, n_o$ ,  $t = 1, 2, \dots$ , is then

$$y_n^t = \sum_{k=1}^{k_o} G(n, k) \cdot x_k^t \oplus \sum_{k=1}^{m_o} G(n, k_o + k) \cdot x_k^{t-1} \quad (1)$$

where  $\sum$  and  $\oplus$  represent the mod-2 summations and  $x_k^t \in (0, 1)$ , for  $k = 1, 2, \dots, k_o$ , and  $t = 1, 2, \dots$  ( $x_k^0 = 0$ ,  $k = 1, 2, \dots, m_o$ , by convention).

First, consider FUM codes with  $m_o = k_o$ . We define the “present state” at time  $t$ ,  $S^t$ , as

$$S^t = (x_1^{t-1}, x_2^{t-1}, \dots, x_{k_o}^{t-1})$$

The “next state” at time  $t$  is then  $S^{t+1}$ . Notice that the number of possible states is  $2^{k_o}$  ( $= M_1$ ), regardless of the number of outputs  $n_o$  and the time  $t$ . Define the state space,  $\mathcal{S}$ , as

$$\mathcal{S} = \{s_0, s_1, \dots, s_{M_1-1}\}$$

where

$$s_i = (x_1, x_2, \dots, x_{k_o})$$

with

$$i = \sum_{k=1}^{k_o} x_k \cdot 2^{k_o-k}$$

for any time  $t$ .

The SD for TFB of this  $(k_o, k_o/n_o)$  FUM code consists of  $M_1$  nodes, and  $M_1^2$  directed branches with associated branch metrics. Node  $i$  represents the state  $s_i$ , and the directed branch from node  $i$  to node  $j$  represents the transition from state  $s_i$  to state  $s_j$ . The branch metric on the directed branch from node  $i$  to node  $j$ ,  $m(i, j)$ , is given by

$$m(i, j) = D^{H^O(i, j)} Z^{H^I(i, j)} \quad (2)$$

where the values of  $H^O$ 's depend on the code generator  $G$  and the type of channel to be used, and the values of  $H^I$ 's depend on the type of performance measure of interest. Here, for the type of channel we will consider only the binary-input symmetric-output (BISO) channel, and for the type of performance measure we will consider the bit error rate (BER) and the  $M_1 (= 2^{k_o})$ -ary symbol error rate (SER). For the BISO channel, the  $H^O$ 's in Eq. (2) are then

$$H^O(i, j) = \sum_{n=1}^{n_o} y_n^t(i, j)$$

where  $y_n^t(i, j)$  is the  $y_n^t$  in Eq. (1) with  $(x_1^{t-1}, x_2^{t-1}, \dots, x_{k_o}^{t-1}) = s_i$  and  $(x_1^t, x_2^t, \dots, x_{k_o}^t) = s_j$ . That is,  $H^O(i, j)$  is the binary Hamming weight of the corresponding output vector. The  $H^I$ 's in Eq. (2) when using the BER criterion,  $H_B^I$ 's, are then

$$H_B^I(i, j) = \frac{(\text{number of 1's in the next state } s_j)}{k_o}$$

i.e.,  $H_B^I(i, j)$  is the normalized binary Hamming weight of the corresponding input vector. Likewise, the  $H^I$ 's in Eq. (2) for  $M_1$ -ary SER,  $H_S^I$ 's, are

$$H_S^I(i, j) = 0, \text{ if } j = 0 \\ = 1, \text{ otherwise}$$

or  $H_S^I(i, j)$  is the  $M_1$ -ary Hamming weight of the corresponding input symbol.

From the SD described, we find the transfer function  $T(D, Z)$  as (Refs. 4 and 5)

$$T(D, Z) = B \cdot (I - A)^{-1} \cdot C$$

where  $I$  is the  $(M_1 - 1) \times (M_1 - 1)$  unit matrix, and the  $(M_1 - 1) \times (M_1 - 1)$  matrix  $A$ , the  $(M_1 - 1)$ -dimensional row vector  $B$ , and the  $(M_1 - 1)$ -dimensional column vector  $C$  are obtained from the SD for  $i = 1, 2, \dots, M_1 - 1$  and  $j = 1, 2, \dots$ ,

$M_1 - 1$ , by  $A(i, j) = m(i, j)$ ,  $B(j) = m(0, j)$ , and  $C(i) = m(i, 0)$ . Then with

$$\begin{aligned} \frac{\partial}{\partial Z} T(D, Z) &= \frac{\partial B}{\partial Z} \cdot (I - A)^{-1} \cdot C \\ &+ B \cdot (I - A)^{-1} \cdot \frac{\partial A}{\partial Z} \cdot (I - A)^{-1} \cdot C \end{aligned}$$

the TFB for BER and SER can be found as

$$\text{BER} \leq c_o \cdot \left. \frac{\partial}{\partial Z} T_B(D, Z) \right|_{D=D_o, Z=1}$$

and

$$\text{SER} \leq c_o \cdot \left. \frac{\partial}{\partial Z} T_S(D, Z) \right|_{D=D_o, Z=1}$$

where  $D_o$  is the union Bhattacharyya distance of the coding channel (everything inside the encoder-decoder pair) and the constant  $c_o$  depends on the type of coding channel and the code. For example, when we use the binary antipodal signaling over an additive white Gaussian noise channel with no channel output quantization,  $D_o$  and  $c_o$  are given by (Ref. 5)

$$D_o = \exp\left(-\frac{E_s}{N_o}\right)$$

and

$$c_o = Q\left(2 \cdot d_f \cdot \frac{E_s}{N_o}\right) \cdot \exp\left(d_f \cdot \frac{E_s}{N_o}\right)$$

where  $N_o$  is the one sided noise spectral density,  $E_s$  is the received signal energy per channel bit,  $d_f$  is the free distance of the code, and

$$Q(w) = \int_w^\infty \exp\left(-\frac{z^2}{2}\right) \cdot \frac{dz}{\sqrt{2\pi}}$$

An illustrating example is shown in Fig. 2. Figure 2(a) shows the encoder structure of a (2, 2/3) FUM code including the code connections. The code generator matrix  $G$  is then

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Its SD when using the BER criterion is shown in Fig. 2(b). From this, we have

$$B = \begin{bmatrix} D & Z^{0.5} & D^3 Z^{0.5} & D^2 Z \end{bmatrix}$$

$$C = \begin{bmatrix} D^2 \\ D^2 \\ D^2 \end{bmatrix}, \quad A = \begin{bmatrix} D & Z^{0.5} & D & Z^{0.5} & D^2 Z \\ D^3 Z^{0.5} & D & Z^{0.5} & Z \\ D & Z^{0.5} & D & Z^{0.5} & D^2 Z \end{bmatrix}$$

And it is easy to show

$$\left. \frac{\partial T_B}{\partial Z} \right|_{Z=1} = \frac{0.5 D^3 + 2 D^4 + 1.5 D^5 - 3 D^6 - D^7 + 4 D^8}{(1 - 3 D + D^3 + D^4)^2}$$

If the SER is the performance measure of interest, then the corresponding SD is exactly the same as Fig. 2(b) but with  $Z^{0.5}$  replaced by  $Z$ . This gives

$$\left. \frac{\partial T_S}{\partial Z} \right|_{Z=1} = \frac{D^3 + 2 D^4 + 3 D^5 - 5 D^6 - 2 D^7 + 5 D^8 - D^{10}}{(1 - 3 D + D^3 + D^4)^2}$$

### III. Reduced State Diagram for TFB of PUM Codes

The transfer function bounding technique described in the previous section is valid when only one transition exists from a given state to another state. In other words, each encoder input requires at least one memory cell. Hence, the performance of an  $(m_o, k_o/n_o)$  PUM code can be found by introducing  $(k_o - m_o)$  dummy binary memory cells and by treating all of that as a  $(k_o, k_o/n_o)$  FUM code. For an illustrating example, a (1, 2/3) PUM code with one dummy memory cell is shown in Fig. 3(a). The corresponding SD of 4 states is shown in Fig. 3(b). From this, we have

$$B = \begin{bmatrix} D^2 Z^{0.5} & D^2 Z^{0.5} & D^2 Z \end{bmatrix}$$

$$C = \begin{bmatrix} 1 \\ D \\ D \end{bmatrix}, \quad A = \begin{bmatrix} D^2 Z^{0.5} & D^2 Z^{0.5} & D^2 Z \\ D^3 Z^{0.5} & D & Z^{0.5} & D Z \\ D^3 Z^{0.5} & D & Z^{0.5} & D Z \end{bmatrix}$$

and

$$\left. \frac{\partial T_B}{\partial Z} \right|_{Z=1} = \frac{0.5 D^2 - 0.5 D^3 + 2 D^4 + 2 D^5 - 4 D^6 + 2 D^8}{(1 - 2 D - D^2 + 2 D^3 - 2 D^5)^2} \quad (3)$$

and

$$\left. \frac{\partial T_S}{\partial Z} \right|_{Z=1} = \frac{D^2 - 2 D^3 + 4 D^4 + 4 D^5 - 8 D^6 + 4 D^8}{(1 - 2 D - D^2 + 2 D^3 - 2 D^5)^2} \quad (4)$$

Notice in Fig. 3(b) that  $m(0, j) = m(1, j)$  and  $m(2, j) = m(3, j)$  for all  $j$ . In general, for any  $(m_o, k_o/n_o)$  PUM code, since there are no connections from the last  $(k_o - m_o)$  dummy binary cells,  $m(i_a, j) = m(i_b, j)$  for any  $j$ , if the first  $m_o$  elements in  $s_{ia}$  are identical to those in  $s_{ib}$ . This observation is the key for reducing the number of necessary states in the SD for the TFB.

Let  $M_2 = 2^{m_o} (\leq M_1/2)$  and  $M_3 = M_1/M_2 = 2^{k_o - m_o} (\geq 2)$ . Since the number of actually working memory cells is  $m_o$ , one may naturally consider that the number of necessary states is  $M_2$ . That is, one may define the "present state" of the PUM code at time  $t$ ,  $\bar{S}^t$ , as

$$\bar{S}^t = (x_1^{t-1}, x_2^{t-1}, \dots, x_{m_o}^{t-1})$$

regardless of the values of  $x_{m_o+1}^{t-1}, \dots, x_{k_o}^{t-1}$ , and the corresponding state space as;

$$\bar{\mathcal{P}} = \{\bar{s}_0, \bar{s}_1, \bar{s}_2, \dots, \bar{s}_{M_2-1}\} \quad (5)$$

where

$$\bar{s}_i = (x_1, x_2, \dots, x_{m_o})$$

with

$$i = \sum_{k=1}^{m_o} x_k \cdot 2^{m_o-k} \quad (6)$$

In other words, one may define  $\bar{s}_i$  instead of defining  $s_{i \cdot M_3}, s_{i \cdot M_3+1}, \dots, s_{i \cdot M_3+M_3-1}$  without distinction. The SD on this state space  $\bar{\mathcal{P}}$  for any purpose has  $M_2$  nodes and  $M_3 \times M_2^2$  directed branches, where node  $i$  represents the state  $\bar{s}_i$  and the  $M_3$  directed branches from node  $i$  to node  $j$  represent the transitions from state  $\bar{s}_i$  to state  $\bar{s}_j$ . In Ref. 2, this state space  $\bar{\mathcal{P}}$  with  $M_2$  states was used for finding free distances of PUM codes.

However, for TFB, we can not use this  $\bar{\mathcal{P}}$  directly. We may use  $\bar{s}_i$ 's for  $i = 1, 2, \dots, M_2 - 1$ , but we have to be careful about  $\bar{s}_0$ . That is, we have to distinguish  $s_i$ 's,  $i = 1, 2, \dots, M_3 - 1$  from  $s_0$ , since nonzero inputs can cause transitions with nonzero outputs from  $s_i$ ,  $i = 0, 1, \dots, M_3 - 1$ , to  $s_j$ ,  $j = 1, 2, \dots, M_3 - 1$ , although the contents of actually working memories are all 0's. Hence, if we represent  $s_0$  by  $\bar{s}_0$ , then we require another state, say  $\bar{s}_{0'}$ , in order to represent  $s_i$ 's,  $i = 1, 2, \dots, M_3 - 1$ . As a result, we require the state space  $\tilde{\mathcal{P}}$  of  $(M_2 + 1)$  states as depicted by

$$\tilde{\mathcal{P}} = \{s_0, s_{0'}, s_1, s_2, \dots, s_{M_2-1}\}$$

where  $\bar{s}_i$ 's are defined as in (5) and (6) for  $i = 1, 2, \dots, M_2 - 1$ , and

$$\bar{s}_0 = (0, 0, \dots, 0) \text{ with } (x_{m_o+1}, \dots, x_{k_o}) = (0, \dots, 0)$$

$$\bar{s}_{0'} = (0, 0, \dots, 0) \text{ with } (x_{m_o+1}, \dots, x_{k_o}) \neq (0, \dots, 0)$$

Now, we can define the reduced SD (RSD) for the TFB of an  $(m_o, k_o/n_o)$  PUM code, which consists of  $(M_2 + 1)$  nodes and  $(M_2 + 1)^2$  directed branches with branch metrics defined for  $i = 0, 1, 2, \dots, M_2 - 1$ , as

$$\bar{m}(i, 0) = m(i \cdot M_3, 0)$$

$$\bar{m}(i, 0') = \sum_{v=1}^{M_3-1} m(i \cdot M_3, v)$$

$$\bar{m}(i, j) = \sum_{v=0}^{M_3-1} m(i \cdot M_3, j \cdot M_3 + v), \quad j = 1, 2, \dots, M_2 - 1$$

and

$$\bar{m}(0', j) = \bar{m}(0, j), \quad j = 0, 0', 1, 2, \dots, M_2 - 1$$

And the corresponding transfer function  $\bar{T}(D, Z)$  is then

$$\bar{T}(D, Z) = \bar{B} \cdot (\bar{I} - \bar{A})^{-1} \cdot \bar{C}$$

where  $\bar{I}$  is the  $M_2 \times M_2$  unit matrix, and for  $i = 0', 1, 2, \dots, M_2 - 1$ , and  $j = 0', 1, 2, \dots, M_2 - 1$ ,  $\bar{A}(i, j) = \bar{m}(i, j)$ ,  $\bar{B}(j) = \bar{m}(0, j)$ , and  $\bar{C}(i) = \bar{m}(i, 0)$ .

For the (1, 2/3) PUM code in Fig. 3(a), the RSD for the TFB using the BER criterion is shown in Fig. 3(c). From this RSD we see that

$$B = \begin{bmatrix} D^2 Z^{0.5} & D^2 Z^{0.5} + D^2 Z \end{bmatrix}$$

$$C = \begin{bmatrix} 1 \\ D \end{bmatrix}, \quad A = \begin{bmatrix} D^2 Z^{0.5} & D^2 Z^{0.5} + D^2 Z \\ D^3 Z^{0.5} & D Z^{0.5} + D Z \end{bmatrix}$$

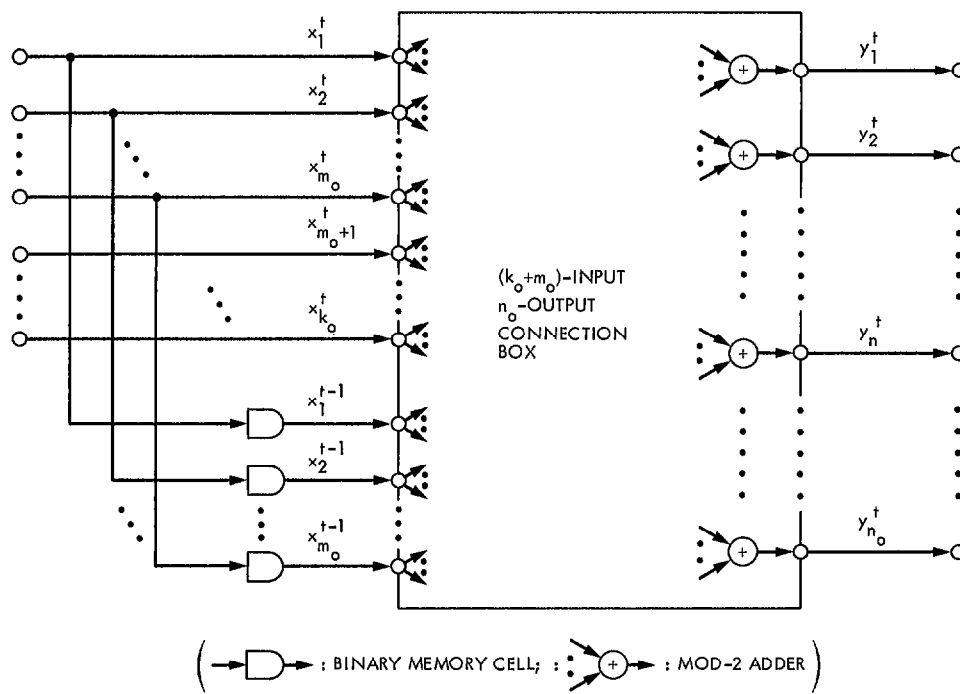
And the resulting expressions for  $(\partial \bar{T}_B / \partial Z)|_{Z=1}$  and  $(\partial \bar{T}_S / \partial Z)|_{Z=1}$  are exactly the same as the right-hand sides of Eqs. (3) and (4).

#### IV. Conclusions and Discussion

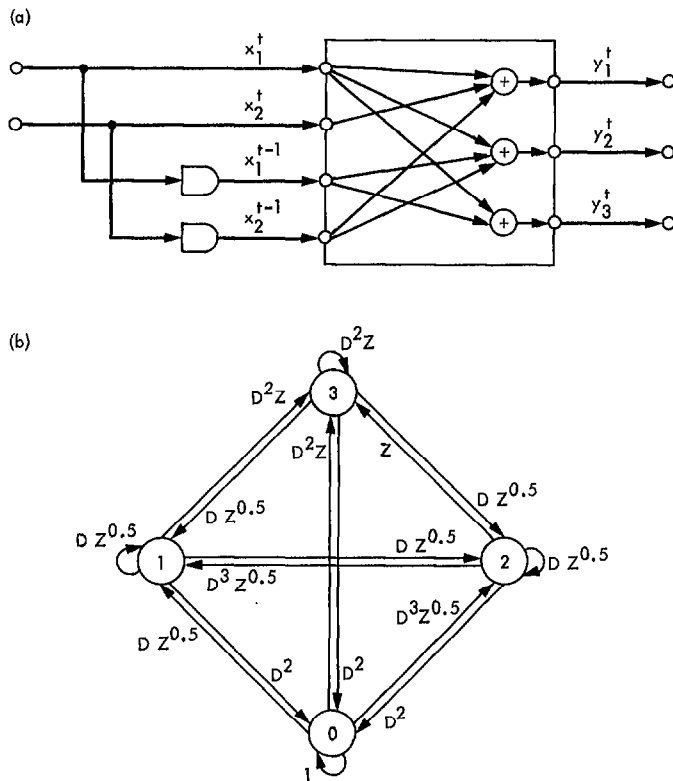
For  $(m_o, k_o/n_o)$  PUM codes, the reduced SD's of  $(2^{m_o} + 1)$  states for TFB's were presented. The TFB's for BER and SER criteria with BISO channel were described. TFB's with other types of channels and/or for other performance criteria may also be of interest. For example, one may want to use  $M(=2^{n_o})$ -ary orthogonal input, symmetric output channel. For another example, one may want to use mean square error for the performance criterion for  $k_o$ -bit representations of certain quantities. For these cases with PUM codes, the corresponding TFB's may also be found based on the reduced SD of  $(2^{m_o} + 1)$  states.

#### References

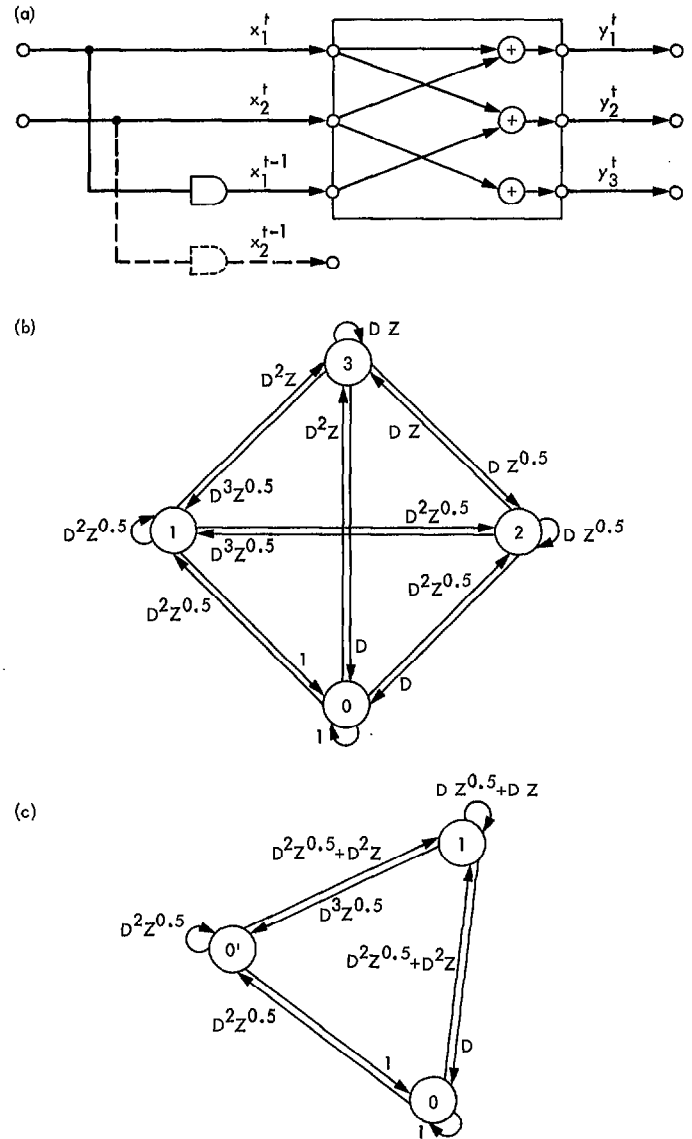
1. Lee, L. N., "Short, Unit-Memory, Byte-Oriented, Binary-Convolutional Codes Having Maximal Free Distance," *IEEE Trans. Info. Theory*, Vol. IT-22, pp. 349-352, May 1976.
2. Lauer, G. S., "Some Optimal Partial-Unit-Memory Codes," *Technical Report No. EE-767*, Univ. of Notre Dame, Notre Dame, Indiana, July 1976. With the same title, its shortened version was published in *IEEE Trans. Info, Theory*, Vol. IT-25, pp. 240-242, March 1979.
3. Lee, L. N., *Concatenated Coding Systems Employing Unit-Memory Convolutional Codes and Symbol-Oriented Optimal Decoding Algorithms*, Ph.D. Thesis, Dept. Elec. Eng., Univ. Notre Dame, Notre Dame, Indiana, May 1976.
4. Omura, J. K., and M. K. Simon, "Modulation/Demodulation Techniques for Satellite Communications, Part IV: Appendix," *JPL Publication 81-73*, Jet Propulsion Laboratory, Pasadena, California, Nov. 1981.
5. Viterbi, A. J., and J. K. Omura, *Principles of Digital Communications and Coding*, McGraw-Hill, New York, 1979.



**Fig. 1. Encoder structure of  $(m_o, k_o/n_o)$  UM code**



**Fig. 2. Example of a (2,2/3) FUM code: (a) encoder structure of a (2,2/3) FUM code; (b) SD of the above code for BER with BISO channel**



**Fig. 3. Example of a (1,2/3) PUM code: (a) encoder structure of a (1,2/3) PUM code; (b) usual SD of the above code for BER with BISO channel; (c) reduced SD of the above code for BER with BISO channel**